```
   CCCCCCCCCCCC     OOOOOOOOO     NNN         NNN  VVV         VVV
   CCCCCCCCCCCC     OOOOOOOOO     NNN         NNN  VVV         VVV
   CCCCCCCCCCCC     OOOOOOOOO     NNN         NNN  VVV         VVV
CCC              OOO         OOO  NNN         NNN  VVV         VVV
CCC              OOO         OOO  NNN         NNN  VVV         VVV
CCC              OOO         OOO  NNN         NNN  VVV         VVV
CCC              OOO         OOO  NNNNNN      NNN  VVV         VVV
CCC              OOO         OOO  NNNNNN      NNN  VVV         VVV
CCC              OOO         OOO  NNNNNN      NNN  VVV         VVV
CCC              OOO         OOO  NNN   NNN   NNN  VVV         VVV
CCC              OOO         OOO  NNN   NNN   NNN  VVV         VVV
CCC              OOO         OOO  NNN   NNN   NNN  VVV         VVV
CCC              OOO         OOO  NNN      NNNNNN  VVV         VVV
CCC              OOO         OOO  NNN      NNNNNN  VVV         VVV
CCC              OOO         OOO  NNN      NNNNNN  VVV         VVV
CCC              OOO         OOO  NNN         NNN     VVV   VVV
CCC              OOO         OOO  NNN         NNN     VVV   VVV
CCC              OOO         OOO  NNN         NNN     VVV   VVV
   CCCCCCCCCCCC     OOOOOOOOO     NNN         NNN        VVV
   CCCCCCCCCCCC     OOOOOOOOO     NNN         NNN        VVV
   CCCCCCCCCCCC     OOOOOOOOO     NNN         NNN        VVV
```

```
**FILE**ID**CONVFILES                                              H 6
```

```
  CCCCCCCC    OOOOOO    NN      NN  VV      VV  FFFFFFFFFF    IIIIII    LL          EEEEEEEEEE    SSSSSSSS
  CCCCCCCC    OOOOOO    NN      NN  VV      VV  FFFFFFFFFF    IIIIII    LL          EEEEEEEEEE    SSSSSSSS
CC          OO      OO  NN      NN  VV      VV  FF              II      LL          EE          SS
CC          OO      OO  NN      NN  VV      VV  FF              II      LL          EE          SS
CC          OO      OO  NNNN    NN  VV      VV  FF              II      LL          EE          SS
CC          OO      OO  NNNN    NN  VV      VV  FF              II      LL          EE          SS
CC          OO      OO  NN  NN  NN  VV      VV  FFFFFFFF        II      LL          EEEEEEEE      SSSSSS
CC          OO      OO  NN  NN  NN  VV      VV  FFFFFFFF        II      LL          EEEEEEEE      SSSSSS
CC          OO      OO  NN    NNNN  VV      VV  FF              II      LL          EE                  SS
CC          OO      OO  NN    NNNN  VV      VV  FF              II      LL          EE                  SS
CC          OO      OO  NN      NN    VV  VV    FF              II      LL          EE                  SS      ....
CC          OO      OO  NN      NN    VV  VV    FF              II      LL          EE                  SS      ....
  CCCCCCCC    OOOOOO    NN      NN      VV      FF            IIIIII    LLLLLLLLLL  EEEEEEEEEE  SSSSSSSS        ....
  CCCCCCCC    OOOOOO    NN      NN      VV      FF            IIIIII    LLLLLLLLLL  EEEEEEEEEE  SSSSSSSS        ....

LL          IIIIII      SSSSSSSS
LL          IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
0001 0 %TITLE  'VAX-11 CONVERT'
0002 0 MODULE  CONV$FILES        ( IDENT='V04-000',
0003 0                          OPTLEVEL=3
0004 0                          ) =
0005 0
0006 1 BEGIN
0007 1
0008 1 !****************************************************************************
0009 1 !*                                                                          *
0010 1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0011 1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0012 1 !*  ALL RIGHTS RESERVED.                                                    *
0013 1 !*                                                                          *
0014 1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0015 1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0016 1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0017 1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0018 1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0019 1 !*  TRANSFERRED.                                                            *
0020 1 !*                                                                          *
0021 1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0022 1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0023 1 !*  CORPORATION.                                                            *
0024 1 !*                                                                          *
0025 1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0026 1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0027 1 !*                                                                          *
0028 1 !*                                                                          *
0029 1 !****************************************************************************
```

```
 1 !++
 1 !
 1 ! Facility:      VAX-11 CONVERT
 1 !
 1 ! Abstract:      RMS file handeling routines
 1 !
 1 ! Contents:
 1 !                PARSE_DEF
 1 !                OPEN_INPUT
 1 !                SEARCH_FILE
 1 !                OPEN_IN
 1 !                OPEN_OUTPUT
 1 !                GET_PROLOGUE
 1 !                CREATE_BUFFER
 1 !
 1 ! Environment:
 1 !
 1 !                VAX/VMS Operating System
 1 !
 1 !
 1 ! Author:        Keith B. Thompson       Creation date:  June-1980
 1 !
 1 !
 1 ! Modified by:
 1 !
 1 !        V03-013 JWT0194         Jim Teague              31-Aug-1984
 1 !                Fix problem with CONVERT dropping blocks when input
 1 !                file is UDF.
 1 !
 1 !        V03-012 RAS0311         Ron Schaefer            18-Jun-1984
 1 !                Fix output file related file parsing by making sure the
 1 !                input file result filespec is available.  Fix to RAS0260.
 1 !
 1 !        V03-011 RAS0272         Ron Schaefer            16-Mar-1984
 1 !                Allow CONVERT to fastload & sort network files since
 1 !                SORT-32 is now abel to handle them.
 1 !
 1 !        V03-010 RAS0260         Ron Schaefer            2-Mar-1984
 1 !                Improve performance of RAS0250 by copying the DVI, FID and
 1 !                DID fields from the LIB$FIND_FILE NAM to the real NAM
 1 !                used for the open.  Also copy the device characteristics.
 1 !
 1 !        V03-009 RAS0250         Ron Schaefer            23-Feb-1984
 1 !                Convert SEARCH_FILE to use LIB$FIND_FILE for correct
 1 !                related file processing.   Add FDL_STRING support.
 1 !
 1 !        V03-008 KBT0442         Keith B. Thompson       30-Dec-1982
 1 !                Make fdl_fab/rab global
 1 !
 1 !        V03-007 KBT0435         Keith B. Thompson       16-Dec-1982
 1 !                Always open the input file to fill the fab
 1 !                except when comming from tape and sorting
 1 !
 1 !        V03-006 KBT0392         Keith B. Thompson       29-Oct-1982
 1 !                Call new read_prologue routine
 1 !
 1 !        V03-005 KBT0370         Keith B. Thompson       19-Oct-1982
```

```
;    88         0087  1 !               Use new supported fdl$parse
;    89         0088  1 !
;    90         0089  1 !       V03-004 KBT0347         Keith B. Thompson       4-Oct-1982
;    91         0090  1 !               Use new linkage definitions
;    92         0091  1 !
;    93         0092  1 !       V03-003 KBT0044         Keith Thompson          5-Apr-1982
;    94         0093  1 !               Don't do a search on a device mounted foreign
;    95         0094  1 !
;    96         0095  1 !       V03-002 KBT0025         Keith Thompson          26-Mar-1982
;    97         0096  1 !               Fix fill switch for /nofast
;    98         0097  1 !
;    99         0098  1 !       V03-001 KBT0015         Keith Thompson          18-Mar-1982
;   100         0099  1 !               Fix area allocation bug in get_prologue and use new
;   101         0100  1 !               plg$c_ver3 instead of literal
;   102         0101  1 !
;   103         0102  1 !****
```

```
1
1 PSECT
1         OWN     = _CONV$OWN       (PIC),
1         GLOBAL  = _CONV$GLOBAL    (PIC),
1         PLIT    = _CONV$PLIT      (SHARE,PIC),
1         CODE    = _CONV$CODE      (SHARE,PIC);
1
1 LIBRARY 'SYS$LIBRARY:LIB.L32';
1 LIBRARY 'SRC$:CONVERT';
1
1 EXTERNAL ROUTINE
1         CONV$$GET_VM              : CL$GET_VM,
1         CONV$$READ_PROLOGUE       : CL$READ_PROLOGUE      NOVALUE,
1         CONV$$RMS_OPEN_ERROR      : NOVALUE,
1         FDL$PARSE                 : ADDRESSING_MODE( GENERAL ),
1         LIB$FIND_FILE             : ADDRESSING_MODE( GENERAL );
1
1 FORWARD ROUTINE
1         CONV$$SEARCH_FILE,
1         CONV$$OPEN_IN;
1
1 !       Error codes
1 !
1 DEFINE_ERROR_CODES;
1
1 EXTERNAL
1
1 !       The Option Flags:
1 !
1         CONV$GL_APPEND            : LONG,                 ! APPEND
1         CONV$GL_CREATE            : LONG,                 ! CREATE
1         CONV$GL_FDL               : LONG,                 ! FDL
1         CONV$GL_EXC               : LONG,                 ! EXCEPTION
1         CONV$GL_FAST              : LONG,                 ! FAST
1         CONV$GL_MERGE             : LONG,                 ! MERGE
1         CONV$GL_FILL              : LONG,                 ! FILL_BUCKETS
1         CONV$GL_FIX               : LONG,                 ! FIXED_WRITE
1         CONV$GL_KEY               : LONG,                 ! KEY
1         CONV$GL_PAD               : LONG,                 ! PAD_RECORDS
1         CONV$GL_SHARE             : LONG,                 ! SHARE
1         CONV$GL_SORT              : LONG,                 ! SORT
1         CONV$GL_READ_C            : LONG,                 ! READ_CHECK
1         CONV$GL_TRUNCATE          : LONG,                 ! TRUNCATE
1         CONV$GL_WRITE_C           : LONG,                 ! WRITE_CHECK
1         CONV$GL_PROLOG            : LONG,                 ! PROLOGUE
1         CONV$AB_FLAGS             : BLOCK [ ,BYTE ],
1
1         CONV$GW_OUT_MRS           : WORD,
1         CONV$GW_UDF_MRS           : WORD,
1         CONV$GB_CURRENT_FILE      : BYTE,
1         CONV$GW_MAX_REC_SIZ       : WORD,
1         CONV$GL_REC_BUF_PTR,
1         CONV$GL_VFC_BUF_PTR,
1         CONV$GL_FINDFILE_CTX,
1
1         CONV$AL_IN_FILE_NAM       : VECTOR [ ,LONG ],             ! Input File
1         CONV$AR_OUT_FILE_NAM      : REF DESC_BLK,                 ! Output File
```

```
1              CONV$AR_FDL_FILE_NAM    : REF DESC_BLK,              ! FDL File
1
1              CONV$AB_IN_XABSUM       : $XABSUM_DECL,
1              CONV$AB_IN_XABFHC       : $XABFHC_DECL,
1              CONV$AB_IN_NAM          : $NAM_DECL,
1              CONV$AB_IN_FAB          : $FAB_DECL,
1              CONV$AB_IN_RAB          : $RAB_DECL,
1              CONV$AB_OUT_XABSUM      : $XABSUM_DECL,
1              CONV$AB_OUT_NAM         : $NAM_DECL,
1              CONV$AB_OUT_FAB         : $FAB_DECL,
1              CONV$AB_OUT_RAB         : $RAB_DECL;
1
1 GLOBAL
1              CONV$AB_FDL_FAB         : REF BLOCK [ ,BYTE ],
1              CONV$AB_FDL_RAB         : REF BLOCK [ ,BYTE ];
1
```

```
1 %SBTTL  'PARSE_DEF'
1 GLOBAL ROUTINE CONV$$PARSE_DEF =
1 !++
1 !
1 ! Functional Description:
1 !
1 !       Calls fdl$parse to parse the fdl file and fill in a fab.  The
1 !       info from this fab is will be copied to the output fab in open_output
1 !
1 ! Calling Sequence:
1 !
1 !       CONV$$PARSE()
1 !
1 ! Input Parameters:
1 !       none
1 !
1 ! Implicit Inputs:
1 !
1 !       CONV$AR_FDL_FILE_NAME   - FDL file descriptor
1 !
1 ! Output Parameters:
1 !       none
1 !
1 ! Implicit Outputs:
1 !       none
1 !
1 ! Routine Value:
1 !
1 !       Value returned by fdl$parse
1 !
1 ! Routines Called:
1 !
1 !       FDL$PARSE
1 !
1 ! Side Effects:
1 !
1 !--
1
2   BEGIN
2
2 !    EXTERNAL LITERAL
2 !       FDL$M_FDL_STRING,
2 !       FDL$M_SIGNAL;
2
2   LOCAL
2       FDL_FLAGS       : LONG;
2
2   ! Initialize the flags
2   !
2   FDL_FLAGS = 0;
2
2   ! If convert is signaling then fdl should
2   !
2   IF .CONV$AB_FLAGS [ CONV$V_SIGNAL ]
2   THEN
2 !     FDL_FLAGS = FDL$M_SIGNAL;
2       FDL_FLAGS = 1;
```

```
	! If caller passed in an fdl string, then tell fdl about it
	!
	IF .CONV$AB_FLAGS [ CONV$V_FDL_STRING ]
	THEN
!	    FDL_FLAGS = FDL$M_FDL_STRING OR .FDL_FLAGS;
	    FDL_FLAGS = 2 OR .FDL_FLAGS;


	RETURN FDL$PARSE( .CONV$AR_FDL_FILE_NAM,
			  CONV$AB_FDL_FAB,
			  CONV$AB_FDL_RAB,
			  FDL_FLAGS )

	END;
```

```
                                        .TITLE  CONV$FILES VAX-11 CONVERT
                                        .IDENT  \V04-000\

                                        .PSECT  _CONV$GLOBAL,NOEXE,  PIC,2

                        00000 CONV$AB_FDL_FAB::
                                        .BLKB   4
                        00004 CONV$AB_FDL_RAB::
                                        .BLKB   4

                                        .EXTRN  CONV$$GET_VM, CONV$$READ_PROLOGUE
                                        .EXTRN  CONV$$RMS_OPEN_ERROR
                                        .EXTRN  FDL$PARSE, LIB$FIND_FILE
                                        .EXTRN  CONVERT$_FACILITY
                                        .EXTRN  CONV$_FAO_MAX, CONV$_BADBLK
                                        .EXTRN  CONV$_BADLOGIC, CONV$_BADSORT
                                        .EXTRN  CONV$_CONFQUAL, CONV$_CREATEDSTM
                                        .EXTRN  CONV$_CREA_ERR, CONV$_DELPRI
                                        .EXTRN  CONV$_DUP, CONV$_EXTN_ERR
                                        .EXTRN  CONV$_FATALEXC, CONV$_FILLIM
                                        .EXTRN  CONV$_IDX_LIM, CONV$_ILL_KEY
                                        .EXTRN  CONV$_ILL_VALUE
                                        .EXTRN  CONV$_INP_FILES
                                        .EXTRN  CONV$_INSVIRMEM
                                        .EXTRN  CONV$_INVBKT, CONV$_KEY
                                        .EXTRN  CONV$_KEYREF, CONV$_LOADIDX
                                        .EXTRN  CONV$_NARG, CONV$_NI
                                        .EXTRN  CONV$_NOKEY, CONV$_NOTIDX
                                        .EXTRN  CONV$_NOTSEQ, CONV$_NOWILD
                                        .EXTRN  CONV$_ORDER, CONV$_OPENEXC
                                        .EXTRN  CONV$_OPENIN, CONV$_OPENOUT
                                        .EXTRN  CONV$_PAD, CONV$_PLV
                                        .EXTRN  CONV$_PROERR, CONV$_PROL_WRT
                                        .EXTRN  CONV$_READERR, CONV$_RSK
                                        .EXTRN  CONV$_RSZ, CONV$_RTL
                                        .EXTRN  CONV$_RTS, CONV$_SEQ
                                        .EXTRN  CONV$_UDF_BKS, CONV$_UDF_BLK
                                        .EXTRN  CONV$_VFC, CONV$_WRITEERR
                                        .EXTRN  CONV$GL_APPEND, CONV$GL_CREATE
                                        .EXTRN  CONV$GL_FDL, CONV$GL_EXC
```

```
                                                    .EXTRN   CONV$GL_FAST, CONV$GL_MERGE
                                                    .EXTRN   CONV$GL_FILL, CONV$GL_FIX
                                                    .EXTRN   CONV$GL_KEY, CONV$GL_PAD
                                                    .EXTRN   CONV$GL_SHARE, CONV$GL_SORT
                                                    .EXTRN   CONV$GL_READ_C, CONV$GL_TRUNCATE
                                                    .EXTRN   CONV$GL_WRITE_C
                                                    .EXTRN   CONV$GL_PROLOG, CONV$AB_FLAGS
                                                    .EXTRN   CONV$GW_OUT_MRS
                                                    .EXTRN   CONV$GW_UDF_MRS
                                                    .EXTRN   CONV$GB_CURRENT_FILE
                                                    .EXTRN   CONV$GW_MAX_REC_SIZ
                                                    .EXTRN   CONV$GL_REC_BUF_PTR
                                                    .EXTRN   CONV$GL_VFC_BUF_PTR
                                                    .EXTRN   CONV$GL_FINDFILE_CTX
                                                    .EXTRN   CONV$AL_IN_FILE_NAM
                                                    .EXTRN   CONV$AR_OUT_FILE_NAM
                                                    .EXTRN   CONV$AR_FDL_FILE_NAM
                                                    .EXTRN   CONV$AB_IN_XABSUM
                                                    .EXTRN   CONV$AB_IN_XABFHC
                                                    .EXTRN   CONV$AB_IN_NAM, CONV$AB_IN_FAB
                                                    .EXTRN   CONV$AB_IN_RAB, CONV$AB_OUT_XABSUM
                                                    .EXTRN   CONV$AB_OUT_NAM
                                                    .EXTRN   CONV$AB_OUT_FAB
                                                    .EXTRN   CONV$AB_OUT_RAB

                                                    .PSECT   _CONV$CODE,NOWRT,  SHR,  PIC,2

                                   0000 00000       .ENTRY   CONV$$PARSE_DEF, Save nothing          ; 0177
                                 7E  D4 00002       CLRL     FDL_FLAGS                              ; 0225
               03        0000G   CF  E9 00004       BLBC     CONV$AB_FLAGS, 1$                      ; 0229
               6E                01  D0 00009       MOVL     #1, FDL_FLAGS                          ; 0232
  03    0000G  CF                01  E1 0000C 1$:   BBC      #1, CONV$AB_FLAGS, 2$                  ; 0236
               6E                02  88 00012       BISB2    #2, FDL_FLAGS                          ; 0239
                                 5E  DD 00015 2$:   PUSHL    SP                                     ; 0242
                         0000'   CF  9F 00017       PUSHAB   CONV$AB_FDL_RAB                        ;
                         0000'   CF  9F 0001B       PUSHAB   CONV$AB_FDL_FAB                        ;
                         0000G   CF  DD 0001F       PUSHL    CONV$AR_FDL_FILE_NAM                   ;
    00000000G  00                04  FB 00023       CALLS    #4, FDL$PARSE                          ;
                                     04 0002A       RET                                             ; 0247

; Routine Size:  43 bytes,    Routine Base:  _CONV$CODE + 0000
```

```
1 %SBTTL  'OPEN_INPUT'
1 GLOBAL ROUTINE  CONV$$OPEN_INPUT =
1 !++
1 !
1 ! Functional Description:
1 !
1 !       Opens an input file
1 !
1 ! Calling Sequence:
1 !
1 !       CONV$$OPEN_INPUT()
1 !
1 ! Input Parameters:
1 !       none
1 !
1 ! Implicit Inputs:
1 !       none
1 !
1 ! Output Parameters:
1 !       none
1 !
1 ! Implicit Outputs:
1 !       none
1 !
1 ! Routine Value:
1 !
1 !       CONV$_SUCCESS or error code from CONV$$SEARCH_FILE or CONV$$OPEN_IN
1 !
1 ! Routines Called:
1 !
1 !       CONV$$SEARCH_FILE
1 !       CONV$$OPEN_IN
1 !
1 ! Side Effects:
1 !
1 !       Opens an input file
1 !
1 !--
1
2     BEGIN
2
2     LOCAL
2         STATUS  : LONG,
2         IN_DEV  : BLOCK [ 1,LONG ];
2
2     ! Any errors on the input fab are OPENIN errors
2     !
2     CONV$AB_IN_FAB [ FAB$L_CTX ] = CONV$_OPENIN;
2
2     ! Start by getting the file name
2     !
2     RET_ON_ERROR( CONV$$SEARCH_FILE() );
2
2     ! For now there are only sequential files on tape if there is no
2     ! definition file then it cant be a fast load
2     !
```

```
;  309        0305  2       IN_DEV = .CONV$AB_IN_FAB [ FAB$L_DEV ];
;  310        0306  2
;  311        0307  2       IF ( NOT .CONV$GL_FDL ) AND .IN_DEV [ DEV$V_SQD ]
;  312        0308  2       THEN
;  313        0309  3           BEGIN
;  314        0310  3           CONV$GL_FAST    = _CLEAR;
;  315        0311  3           CONV$GL_SORT    = _CLEAR
;  316        0312  2           END;
;  317        0313  2
;  318        0314  2       ! If we are sorting the input file from tape or network
;  319        0315  2       ! then don't bother opening the input file here
;  320        0316  2       !
;  321        0317  2       IF .CONV$GL_SORT AND (.IN_DEV [ DEV$V_SQD ] OR .IN_DEV [ DEV$V_NET ])
;  322        0318  2       THEN
;  323        0319  2           RETURN CONV$_SUCCESS
;  324        0320  2       ELSE
;  325        0321  2           RETURN CONV$$OPEN_IN()
;  326        0322  2
;  327        0323  1       END;



                                    0000 00000          .ENTRY  CONV$$OPEN_INPUT, Save nothing                 ; 0249
        0000G  CF 00000000G  8F  D0 00002          MOVL    #CONV$_OPENIN, CONV$AB_IN_FAB+24                ; 0296
        0000V  CF            00  FB 0000B          CALLS   #0, CONV$$SEARCH_FILE                           ; 0300
               2C            50  E9 00010          BLBC    STATUS, 4$                                      ;
               50     0000G  CF  D0 00013          MOVL    CONV$AB_IN_FAB+64, IN_DEV                       ; 0305
               0C     0000G  CF  E8 00018          BLBS    CONV$GL_FDL, 1$                                 ; 0307
  08           50            05  E1 0001D          BBC     #5, IN_DEV, 1$                                  ;
                      0000G  CF  D4 00021          CLRL    CONV$GL_FAST                                    ; 0310
                      0000G  CF  D4 00025          CLRL    CONV$GL_SORT                                    ; 0311
               0C     0000G  CF  E9 00029 1$:      BLBC    CONV$GL_SORT, 3$                                ; 0317
  04           50            05  E0 0002E          BBS     #5, IN_DEV, 2$                                  ;
  04           50            0D  E1 00032          BBC     #13, IN_DEV, 3$                                 ;
               50            01  D0 00036 2$:      MOVL    #1, R0                                          ; 0321
                                 04 00039          RET                                                     ;
        0000V  CF            00  FB 0003A 3$:      CALLS   #0, CONV$$OPEN_IN                               ;
                                 04 0003F 4$:      RET                                                     ; 0323

; Routine Size:  64 bytes,    Routine Base:  _CONV$CODE + 002B
```

```
1 %SBTTL  'SEARCH_FILE'
1 GLOBAL ROUTINE CONV$$SEARCH_FILE =
1 !++
1 !
1 ! Functional Description:
1 !
1 !       Searches for an input file
1 !
1 ! Calling Sequence:
1 !
1 !       CONV$$SEARCH_FILE()
1 !
1 ! Input Parameters:
1 !       none
1 !
1 ! Implicit Inputs:
1 !
1 !       CONV$AL_IN_FILE_NAME    - Input file name descriptor array
1 !       CONV$GB_CURRENT_FILE    - Current input file name being searched
1 !       CONV$GL_FINDFILE_CTX    - LIB$FIND_FILE context
1 !
1 ! Output Parameters:
1 !       none
1 !
1 ! Implicit Outputs:
1 !       none
1 !
1 ! Routine Value:
1 !
1 !       CONV$_SUCCESS or CONV$_NOWILD
1 !
1 ! Routines Called:
1 !
1 !       LIB$FIND_FILE
1 !       $PARSE
1 !       CONV$$RMS_OPEN_ERROR    - By RMS as an AST or by us on LIB$FIND_FILE errors
1 !       $SEARCH
1 !
1 ! Side Effects:
1 !
1 !       Sets up input name block for next input file
1 !
1 !--
1
2     BEGIN
2
2     LOCAL
2         STATUS,
2         STV,
2         FINDFILENAM     : REF BLOCK [ ,BYTE ],
2         IN_NAME         : REF DESC_BLK,
2         OUT_NAME        : DESC_BLK,
2         IN_DEVICE       : BLOCK [ 1,LONG ];
2
2     BIND
2         FINDFILEFAB     = CONV$GL_FINDFILE_CTX : REF BLOCK[ ,BYTE];
2
```

```
;  386        0381  2        IN_NAME = .CONV$AL_IN_FILE_NAM [ .CONV$GB_CURRENT_FILE ];
;  387        0382  2
;  388        0383  2        OUT_NAME [DSC$B_CLASS] = DSC$K_CLASS_D;
;  389        0384  2        OUT_NAME [DSC$B_DTYPE] = DSC$K_DTYPE_T;
;  390        0385  2        OUT_NAME [DSC$W_LENGTH] = 0;
;  391        0386  2        OUT_NAME [DSC$A_POINTER] = 0;
;  392        0387  2
;  393        0388  2        ! Get the next file name to search for
;  394        0389  2        !
;  395        0390  2        STATUS = LIB$FIND_FILE(
;  396        0391  2                    .IN_NAME, OUT_NAME,
;  397        0392  2                    CONV$GL_FINDFILE_CTX,
;  398        0393  2                    0, 0, STV, %REF(3));
;  399        0394  2
;  400        0395  2        ! If the filename has wildcards in it it's an error
;  401        0396  2        !
;  402        0397  2        IF (.STATUS AND STS$M_MSG_NO)  EQL  SHR$_NOWILD
;  403        0398  2        THEN
;  404        0399  2            RETURN CONV$_NOWILD;
;  405        0400  2
;  406        0401  2        ! Report miscellaneous errors from LIB$FIND_FILE
;  407        0402  2        !
;  408        0403  2        IF NOT .STATUS
;  409        0404  2        THEN
;  410        0405  3            BEGIN
;  411        0406  3            FINDFILEFAB [ FAB$L_CTX ] = CONV$_OPENIN;
;  412        0407  3            CONV$$RMS_OPEN_ERROR(.FINDFILEFAB);
;  413        0408  2            END;
;  414        0409  2
;  415        0410  2        CONV$AB_IN_FAB [ FAB$B_FNS ] = .OUT_NAME [ DSC$W_LENGTH ];
;  416        0411  2        CONV$AB_IN_FAB [ FAB$L_FNA ] = .OUT_NAME [ DSC$A_POINTER ];
;  417        0412  2
;  418        0413  2        ! Clear the IFI and device char. so we can parse
;  419        0414  2        !
;  420        0415  2        CONV$AB_IN_FAB [ FAB$W_IFI ] = 0;
;  421        0416  2        CONV$AB_IN_FAB [ FAB$L_DEV ] = .FINDFILEFAB [ FAB$L_DEV ];
;  422        0417  2
;  423        0418  2        FINDFILENAM = .FINDFILEFAB [ FAB$L_NAM ];
;  424        0419  2
;  425        0420  2        ! Copy the DVI, FID and DID fields to the NAM block to use for the open.
;  426        0421  2        !
;  427        0422  2        CH$MOVE( NAM$S_DVI+NAM$S_FID+NAM$S_DID,
;  428        0423  2            FINDFILENAM [ NAM$T_DVI ], CONV$AB_IN_NAM [ NAM$T_DVI ]);
;  429        0424  2
;  430        0425  2        RETURN CONV$_SUCCESS
;  431        0426  2
;  432        0427  1        END;



                                      007C 00000          .ENTRY   CONV$$SEARCH_FILE, Save R2,R3,R4,R5,R6          : 0325
                   56      0000G   CF   9E 00002          MOVAB    CONV$GL_FINDFILE_CTX, R6                        :
                   5E              0C   C2 00007          SUBL2    #12, SP                                         :
                   50      0000G   CF   9A 0000A          MOVZBL   CONV$GB_CURRENT_FILE, R0                        : 0381
                   50      0000GCF40    D0 0000F          MOVL     CONV$AL_IN_FILE_NAM[R0], IN_NAME                :
                       04  AE 020E0000  8F  D0 00015        MOVL    #34471936, OUT_NAME                           ; 0385
                                   08  AE  D4 0001D        CLRL    OUT_NAME+4                                    ; 0386
                                       03  DD 00020        PUSHL   #3                                            ; 0393
                                       5E  DD 00022        PUSHL   SP                                            ;
                                   08  AE  9F 00024        PUSHAB  STV                                           ; 0390
                                       7E  7C 00027        CLRQ    -(SP)                                         ;
                                       56  DD 00029        PUSHL   R6                                            ;
                                   1C  AE  9F 0002B        PUSHAB  OUT_NAME                                      ;
                                       50  DD 0002E        PUSHL   IN_NAME                                       ; 0391
                  00000000G  00        07  FB 00030        CALLS   #7, LIB$FIND_FILE                             ;
                             51        50  D0 00037        MOVL    R0, STATUS                                    ;
              50             51 FFFF0007 8F CB 0003A       BICL3   #-65529, STATUS, R0                           ; 0397
                  00001128   8F        50  D1 00042        CMPL    R0, #4392                                     ;
                                       08  12 00049        BNEQ    1$                                            ;
                             50 00000000G 8F D0 0004B      MOVL    #CONV$_NOWILD, R0                             ; 0399
                                           04 00052        RET                                                   ;
                             12        51  E8 00053  1$:   BLBS    STATUS, 2$                                    ; 0403
                             50        66  D0 00056        MOVL    FINDFILEFAB, R0                               ; 0406
                       18    A0 00000000G 8F D0 00059      MOVL    #CONV$_OPENIN, 24(R0)                         ;
                                       50  DD 00061        PUSHL   R0                                            ; 0407
                     0000G   CF        01  FB 00063        CALLS   #1, CONV$$RMS_OPEN_ERROR                      ;
                     0000G   CF     08  AE 90 00068  2$:   MOVB    OUT_NAME, CONV$AB_IN_FAB+52                   ; 0410
                     0000G   CF     0C  AE D0 0006E        MOVL    OUT_NAME+4, CONV$AB_IN_FAB+44                 ; 0411
                               0000G  CF B4 00074          CLRW    CONV$AB_IN_FAB+2                              ; 0415
                             50        66  D0 00078        MOVL    FINDFILEFAB, R0                               ; 0416
                     0000G   CF     40  A0 D0 0007B        MOVL    64(R0), CONV$AB_IN_FAB+64                     ;
                             50     28  A0 D0 00081        MOVL    40(R0), FINDFILENAM                           ; 0418
            0000G   CF       14  A0    1C  28 00085        MOVC3   #28, 20(FINDFILENAM), CONV$AB_IN_NAM+20      ; 0423
                             50        01  D0 0008C        MOVL    #1, R0                                        ; 0425
                                           04 0008F        RET                                                   ; 0427

; Routine Size:  144 bytes,    Routine Base:  _CONV$CODE + 006B


;  433         0428  1
```

```
1 %SBTTL  'OPEN_IN'
1 GLOBAL ROUTINE  CONV$$OPEN_IN =
1 !++
1 !
1 ! Functional Description:
1 !
1 !       Actually does the open of the input file, allocates and fills
1 !       in key and area xabs if necessary, also connects record stream
1 !
1 ! Calling Sequence:
1 !
1 !       CONV$$OPEN_IN()
1 !
1 ! Input Parameters:
1 !       none
1 !
1 ! Implicit Inputs:
1 !
1 !       CONV$AB_IN_FAB  - Input fab
1 !
1 ! Output Parameters:
1 !       none
1 !
1 ! Implicit Outputs:
1 !       none
1 !
1 ! Routine Value:
1 !
1 !       CONV$_SUCCESS or CONV$_NOKEY
1 !
1 ! Routines Called:
1 !
1 !       $OPEN
1 !       CONV$$RMS_OPEN_ERROR    - By RMS as an AST
1 !       CONV$$GET_VM
1 !       $DISPLAY
1 !       $CONNECT
1 !
1 ! Side Effects:
1 !
1 !       Opens and connects input file
1 !
1 !--
1
2     BEGIN
2
2     LOCAL
2         STATUS  : LONG;
2
2     ! Set the FAB from the Option Switches
2     !
2     ! Read Check
2     !
2     CONV$AB_IN_FAB [ FAB$V_RCK ] = .CONV$GL_READ_C;
2
2     ! Input file sharing
2     !
```

```
2       IF .CONV$GL_SHARE
2       THEN
3           BEGIN
3
3           ! Set up the file sharing bits
3           !
3           CONV$AB_IN_FAB [ FAB$B_SHR ] = FAB$M_PUT OR FAB$M_GET OR FAB$M_DEL OR
3                                                   FAB$M_UPD OR FAB$M_UPI;
3
3           ! Do not wait for any record locks
3           !
3           CONV$AB_IN_RAB [ RAB$V_RRL ] = _SET
3
3           END;
2
2       ! If we have to access the file by key (other then primary) or we have
2       ! to sort the file (which means we use RFA access)
2       ! then clear the sqo bit
2       !
2       IF ( .CONV$GL_KEY NEQU 0 ) OR .CONV$GL_SORT
2       THEN
2           CONV$AB_IN_FAB [ FAB$V_SQO ] = _CLEAR;
2
2       ! Open the file
2       !
2       $OPEN ( FAB=CONV$AB_IN_FAB,ERR=CONV$$RMS_OPEN_ERROR );
2
2       ! Say that the file is open
2       !
2       CONV$AB_FLAGS [ CONV$V_IN ] = _SET;
2
2       ! If this is an index file and we are creating the output file not by
2       ! FDL definition then get the area ad key xabs
2       !
2       IF ( .CONV$AB_IN_FAB [ FAB$B_ORG ] EQLU FAB$C_IDX ) AND
3                               .CONV$GL_CREATE AND ( NOT .CONV$GL_FDL )
2       THEN
3           BEGIN
3
3           LOCAL
3                   BYTES,
3                   VM_POINTER,
3                   CURRENTXAB      : REF BLOCK [ ,BYTE ];
3
3           BIND
3                   NEWXAB  = VM_POINTER : REF BLOCK [ ,BYTE ];
3
3           ! Find out how much memory we need (The extra 32 is for the key name buffer)
3           !
3           BYTES = .CONV$AB_IN_XABSUM [ XAB$B_NOK ] * ( XAB$C_KEYLEN + 32 );
3           BYTES = ( .CONV$AB_IN_XABSUM [ XAB$B_NOA ] * XAB$C_ALLLEN ) + .BYTES;
3
3           ! Get the address space
3           !
3           VM_POINTER = CONV$$GET_VM ( .BYTES  );
3
3           ! The protection xab will point to the new xabs
```

```
3                 !
3                 CURRENTXAB = CONV$AB_IN_XABSUM;
3
3                 ! Chain the xabs together and set up the fields
3                 ! Keys first
3                 !
3                 INCR I FROM 0 TO .CONV$AB_IN_XABSUM [ XAB$B_NOK ] - 1 BY 1
3                 DO
4                     BEGIN
4                     CURRENTXAB [ XAB$L_NXT ] = .NEWXAB;
4                     CURRENTXAB = .NEWXAB;
4                     CURRENTXAB [ XAB$B_COD ] = XAB$C_KEY;
4                     CURRENTXAB [ XAB$B_BLN ] = XAB$C_KEYLEN;
4                     CURRENTXAB [ XAB$B_REF ] = .I;
4                     CURRENTXAB [ XAB$L_KNM ] = .CURRENTXAB + XAB$C_KEYLEN;
4                     NEWXAB = .NEWXAB + XAB$C_KEYLEN + 32
3                     END;
3
3                 ! Then areas
3                 !
3                 INCR I FROM 0 TO .CONV$AB_IN_XABSUM [ XAB$B_NOA ] - 1 BY 1
3                 DO
4                     BEGIN
4                     CURRENTXAB [ XAB$L_NXT ] = .NEWXAB;
4                     CURRENTXAB = .NEWXAB;
4                     CURRENTXAB [ XAB$B_COD ] = XAB$C_ALL;
4                     CURRENTXAB [ XAB$B_BLN ] = XAB$C_ALLLEN;
4                     CURRENTXAB [ XAB$B_AID ] = .I;
4                     NEWXAB = .NEWXAB + XAB$C_ALLLEN
3                     END;
3
3                 ! The last xab points to 0
3                 !
3                 CURRENTXAB [ XAB$L_NXT ] = 0;
3
3                 ! Do a display to fill it all in
3                 !
4                 $DISPLAY ( FAB=CONV$AB_IN_FAB )
4
2                 END;
2
2             ! If this is an indexed file then set the key of ref. to input on
2             !
2             IF .CONV$AB_IN_FAB [ FAB$B_ORG ] EQL FAB$C_IDX
2             THEN
2
2                 ! If the key of ref. is out of range then signal an error and return
2                 ! normal. (so we can continue)
2                 !
2                 IF .CONV$GL_KEY GEQ .CONV$AB_IN_XABSUM [ XAB$B_NOK ]
2                 THEN
2                     RETURN CONV$_NOKEY
2                 ELSE
2                     CONV$AB_IN_RAB [ RAB$B_KRF ] = .CONV$GL_KEY;
2
2             ! Must Special Case for a UDF (Undefined) Input File
2             !
```

```
;  606          0600  2        IF .CONV$AB_IN_FAB [ FAB$B_RFM ] EQL FAB$C_UDF
;  607          0601  2        THEN
;  608          0602  3            BEGIN
;  609          0603  3
;  610          0604  3            ! Get ready to input the file with Block IO
;  611          0605  3            !
;  612          0606  3            CONV$AB_IN_RAB [ RAB$L_BKT ] = 0;
;  613          0607  3            CONV$AB_IN_RAB [ RAB$V_BIO ] = _SET
;  614          0608  3            END
;  615          0609  2        ELSE
;  616          0610  2
;  617          0611  2            ! Else do normal record IO
;  618          0612  2            !
;  619          0613  2            CONV$AB_IN_RAB [ RAB$V_BIO ] = _CLEAR;
;  620          0614  2
;  621          0615  2        ! In normal operation IN_RAB points to IN_FAB but may be changed
;  622          0616  2        ! when doing sorts
;  623          0617  2        !
;  624          0618  2        CONV$AB_IN_RAB [ RAB$L_FAB ] = CONV$AB_IN_FAB;
;  625          0619  2
;  626          0620  2        ! Now that every thing is ready connect a stream
;  627          0621  2        !
;  628          0622  2        $CONNECT ( RAB=CONV$AB_IN_RAB,ERR=CONV$$RMS_OPEN_ERROR );
;  629          0623  2
;  630          0624  2        ! Any errors from now on are read errors
;  631          0625  2        !
;  632          0626  2        CONV$AB_IN_RAB [ RAB$L_CTX ] = CONV$_READERR;
;  633          0627  2
;  634          0628  2        RETURN CONV$_SUCCESS
;  635          0629  2
;  636          0630  1        END;


                                                                        .EXTRN  SYS$OPEN, SYS$DISPLAY
                                                                        .EXTRN  SYS$CONNECT

                                                     0FFC 00000         .ENTRY  CONV$$OPEN_IN, Save R2,R3,R4,R5,R6,R7,R8,-    ; 0430
                                                                                R9,R10,R11                                    ;
                                  57       0000G  CF   9E 00002         MOVAB   CONV$GL_KEY, R7                               ;
                                  56       0000G  CF   9E 00007         MOVAB   CONV$AB_IN_XABSUM+9, R6                       ;
                                  55       0000G  CF   9E 0000C         MOVAB   CONV$AB_IN_RAB+4, R5                          ;
                                  54       0000G  CF   9E 00011         MOVAB   CONV$AB_IN_FAB, R4                            ;
      06   A4            01       07       0000G  CF   F0 00016         INSV    CONV$GL_READ_C, #7, #1, CONV$AB_IN_FAB+6      ; 0482
                                  08       0000G  CF   E9 0001E         BLBC    CONV$GL_SHARE, 1$                             ; 0486
                        17        A4          4F  8F   90 00023         MOVB    #79, CONV$AB_IN_FAB+23                        ; 0493
                                  65              08   88 00028         BISB2   #8, CONV$AB_IN_RAB+4                          ; 0497
                                                  67   D5 0002B 1$:     TSTL    CONV$GL_KEY                                   ; 0505
                                                  05   12 0002D         BNEQ    2$                                            ;
                                  05       0000G  CF   E9 0002F         BLBC    CONV$GL_SORT, 3$                              ;
                        04        A4          40  8F   8A 00034 2$:     BICB2   #64, CONV$AB_IN_FAB+4                         ; 0507
                                           0000G  CF   9F 00039 3$:     PUSHAB  CONV$$RMS_OPEN_ERROR                          ; 0511
                                                  54   DD 0003D         PUSHL   R4                                            ;
                     00000000G    00              02   FB 0003F         CALLS   #2, SYS$OPEN                                  ;
                         0000G    CF              01   88 00046         BISB2   #1, CONV$AB_FLAGS+2                           ; 0515
                                  20          1D  A4   91 0004B         CMPB    CONV$AB_IN_FAB+29, #32                        ; 0520
                                                  7B   12 0004F         BNEQ    8$                                            ;
```

```
                                 76      0000G     CF  E9 00051        BLBC    CONV$GL_CREATE, 8$                                    ; 0521
                                 71      0000G     CF  E8 00056        BLBS    CONV$GL_FDL, 8$                                       ;
                                 51                66  9A 0005B        MOVZBL  CONV$AB_IN_XABSUM+9, BYTES                            ; 0535
                                 51  0000006C      8F  C4 0005E        MULL2   #108, BYTES                                           ;
                                 50         FF     A6  9A 00065        MOVZBL  CONV$AB_IN_XABSUM+8, R0                               ; 0536
                                 50                20  C4 00069        MULL2   #32, R0                                               ;
                                 51                50  C0 0006C        ADDL2   R0, BYTES                                             ;
                                                   51  DD 0006F        PUSHL   BYTES                                                 ; 0540
                                                0000G  30 00071        BSBW    CONV$$GET_VM                                          ;
                                 5E                04  C0 00074        ADDL2   #4, SP                                                ;
                                 51         F7     A6  9E 00077        MOVAB   CONV$AB_IN_XABSUM, CURRENTXAB                         ; 0544
                                 53                66  9A 0007B        MOVZBL  CONV$AB_IN_XABSUM+9, R3                               ; 0549
                                 52                01  CE 0007E        MNEGL   #1, I                                                 ;
                                                   19  11 00081        BRB     5$                                                    ;
                            04   A1                50  D0 00083 4$:    MOVL    NEWXAB, 4(CURRENTXAB)                                 ; 0552
                                 51                80  7E 00087        MOVAQ   (NEWXAB)+, CURRENTXAB                                 ; 0553
                                 61       4C15     8F  B0 0008A        MOVW    #19477, (CURRENTXAB)                                  ; 0554
                            17   A1                52  90 0008F        MOVB    I, 23(CURRENTXAB)                                     ; 0556
                            38   A1         4C     A1  9E 00093        MOVAB   76(R1), 56(CURRENTXAB)                                ; 0557
                                 50         64     A0  9E 00098        MOVAB   100(R0), NEWXAB                                       ; 0558
                   E3            52                53  F2 0009C 5$:    AOBLSS  R3, I, 4$                                             ;
                                 53         FF     A6  9A 000A0        MOVZBL  CONV$AB_IN_XABSUM+8, R3                               ; 0563
                                 52                01  CE 000A4        MNEGL   #1, I                                                 ;
                                                   13  11 000A7        BRB     7$                                                    ;
                            04   A1                50  D0 000A9 6$:    MOVL    NEWXAB, 4(CURRENTXAB)                                 ; 0566
                                 51                80  7E 000AD        MOVAQ   (NEWXAB)+, CURRENTXAB                                 ; 0567
                                 61       2014     8F  B0 000B0        MOVW    #8212, (CURRENTXAB)                                   ; 0568
                            17   A1                52  90 000B5        MOVB    I, 23(CURRENTXAB)                                     ; 0570
                                 50                18  C0 000B9        ADDL2   #24, NEWXAB                                           ; 0571
                   E9            52                53  F2 000BC 7$:    AOBLSS  R3, I, 6$                                             ;
                                            04     A1  D4 000C0        CLRL    4(CURRENTXAB)                                         ; 0576
                                                   54  DD 000C3        PUSHL   R4                                                    ; 0580
                     00000000G   00                01  FB 000C5        CALLS   #1, SYS$DISPLAY                                       ;
                                 20         1D     A4  91 000CC 8$:    CMPB    CONV$AB_IN_FAB+29, #32                                ; 0586
                                                   13  12 000D0        BNEQ    10$                                                   ;
          67       66            08                00  ED 000D2        CMPZV   #0, #8, CONV$AB_IN_XABSUM+9, CONV$GL_KEY             ; 0592
                                                   08  14 000D7        BGTR    9$                                                    ;
                                 50  00000000G     8F  D0 000D9        MOVL    #CONV$_NOKEY, R0                                      ; 0594
                                                       04 000E0        RET                                                           ;
                            31   A5                67  90 000E1 9$:    MOVB    CONV$GL_KEY, CONV$AB_IN_RAB+53                        ; 0596
                                            1F     A4  95 000E5 10$:   TSTB    CONV$AB_IN_FAB+31                                     ; 0600
                                                   09  12 000E8        BNEQ    11$                                                   ;
                                            34     A5  D4 000EA        CLRL    CONV$AB_IN_RAB+56                                     ; 0606
                            01   A5                08  88 000ED        BISB2   #8, CONV$AB_IN_RAB+5                                  ; 0607
                                                   04  11 000F1        BRB     12$                                                   ;
                            01   A5                08  8A 000F3 11$:   BICB2   #8, CONV$AB_IN_RAB+5                                  ; 0613
                            38   A5                64  9E 000F7 12$:   MOVAB   CONV$AB_IN_FAB, CONV$AB_IN_RAB+60                     ; 0618
                                          0000G    CF  9F 000FB        PUSHAB  CONV$$RMS_OPEN_ERROR                                  ; 0622
                                            FC     A5  9F 000FF        PUSHAB  CONV$AB_IN_RAB                                        ;
                     00000000G   00                02  FB 00102        CALLS   #2, SYS$CONNECT                                       ;
                            14   A5  00000000G     8F  D0 00109        MOVL    #CONV$_READERR, CONV$AB_IN_RAB+24                     ; 0626
                                 50                01  D0 00111        MOVL    #1, R0                                                ; 0628
                                                       04 00114        RET                                                           ; 0630

; Routine Size:  277 bytes,    Routine Base:  _CONV$CODE + 00FB
```

```
1 %SBTTL  'OPEN_OUTPUT'
1 GLOBAL ROUTINE  CONV$$OPEN_OUTPUT =
1 !++
1 !
1 ! Functional Description:
1 !
1 !       Creates ( or opens ) the output file, connects a record stream and if
1 !       it is an indexed file allocates and fills in the prologue key and
1 !       area descriptor blocks for sort and/or fast load
1 !
1 ! Calling Sequence:
1 !
1 !       CONV$$OPEN_OUTPUT
1 !
1 ! Input Parameters:
1 !       none
1 !
1 ! Implicit Inputs:
1 !
1 !       CONV$AB_OUT_FAB - Output fab
1 !       CONV$AB_IN_FAB  - Input fab
1 !       Option flags
1 !
1 ! Output Parameters:
1 !       none
1 !
1 ! Implicit Outputs:
1 !
1 !       CONV$AB_FLAGS [ CONV$V_OUT ]    - Set on success
1 !
1 ! Routine Value:
1 !
1 !       CONV$_SUCCESS or error from CONV$$OPEN_IN
1 !
1 ! Routines Called:
1 !
1 !       $PARSE
1 !       CONV$$RMS_OPEN_ERROR    - By RMS as an AST
1 !       $CREATE
1 !       $DISPLAY
1 !       $OPEN
1 !       CONV$$READ_PROLOGUE
1 !       $CONNECT
1 !       CONV$$OPEN_IN
1 !
1 ! Side Effects:
1 !       none
1 !
1 !--
1
2     BEGIN
2
2     LOCAL
2       PRESENT : LONG,
2       OUT_DEV : BLOCK [ 1,LONG ],
2       STATUS  : LONG;
2
```

```
2       ! Any rms errors on the output fab are OPENOUT errors
2       !
2       CONV$AB_OUT_FAB [ FAB$L_CTX ] = CONV$_OPENOUT;
2
2       ! Use the file name in the call argument (not one from FDL)
2       !
2       CONV$AB_OUT_FAB [ FAB$B_FNS ] = .CONV$AR_OUT_FILE_NAM [ DSC$W_LENGTH ];
2       CONV$AB_OUT_FAB [ FAB$L_FNA ] = .CONV$AR_OUT_FILE_NAM [ DSC$A_POINTER ];
2
2       ! Setup the input file name for output file related file parsing
2       !
2       CONV$AB_IN_NAM [ NAM$B_RSL ] = .CONV$AB_IN_FAB [ FAB$B_FNS ];
2       CONV$AB_IN_NAM [ NAM$L_RSA ] = .CONV$AB_IN_FAB [ FAB$L_FNA ];
2
2       ! Parse the name
2       !
2       $PARSE ( FAB=CONV$AB_OUT_FAB,ERR=CONV$$RMS_OPEN_ERROR );
2
2       ! Check the device that the output file is on
2       !
2       OUT_DEV = .CONV$AB_OUT_FAB [ FAB$L_DEV ];
2
2       ! We cannot fast load To a network device
2       !
2       IF .OUT_DEV [ DEV$V_NET ]
2       THEN
2           CONV$GL_FAST = _CLEAR;
2
2       ! Set the FAB from the Option Switches
2       !
2       ! Write Check
2       !
2       CONV$AB_OUT_FAB [ FAB$V_WCK ] = .CONV$GL_WRITE_C;
2
2       ! Merge
2       !
2       CONV$AB_OUT_FAB [ FAB$V_SQO ] = ( NOT .CONV$GL_MERGE );
2
2       ! If the CREATE Option was specified then Create the output file
2       ! else just open it
2       !
2       IF .CONV$GL_CREATE
2       THEN
3           BEGIN
3
3           LOCAL COPY_FAB  : REF BLOCK [ ,BYTE ];
3
3           ! Determine where to copy fab from
3           !
3           IF .CONV$GL_FDL
3           THEN
4               BEGIN
4
4               ! If fdl was done copy the stuff from the fab produced by
4               ! fdl$$parse
4               !
4               COPY_FAB = .CONV$AB_FDL_FAB;
```

```
		    ! Connect the fdl xabs
		    !
		    CONV$AB_OUT_XABSUM [ XAB$L_NXT ] = .COPY_FAB [ FAB$L_XAB ]

		    END
		ELSE
		    BEGIN

		    ! If this is not a create by FDL definition then get the stuff
		    ! from the input file
		    !
		    COPY_FAB = CONV$AB_IN_FAB;

		    ! Connect the input files summary xab NXT which will connect
		    ! any other xabs that the input file may have had ie. area and
		    ! key xabs
		    !
		    CONV$AB_OUT_XABSUM [ XAB$L_NXT ] = .CONV$AB_IN_XABSUM [ XAB$L_NXT ]

		    END;

		! Copy the important fab fields
		!
		CONV$AB_OUT_FAB [ FAB$L_ALQ ] = .COPY_FAB [ FAB$L_ALQ ];	! Allocation
		CONV$AB_OUT_FAB [ FAB$W_DEQ ] = .COPY_FAB [ FAB$W_DEQ ];	! Extension
		CONV$AB_OUT_FAB [ FAB$B_RTV ] = .COPY_FAB [ FAB$B_RTV ];	! Reteval vindoow
		CONV$AB_OUT_FAB [ FAB$B_ORG ] = .COPY_FAB [ FAB$B_ORG ];	! Organization
		CONV$AB_OUT_FAB [ FAB$B_RAT ] = .COPY_FAB [ FAB$B_RAT ];	! Record attributes
		CONV$AB_OUT_FAB [ FAB$B_RFM ] = .COPY_FAB [ FAB$B_RFM ];	! Record format
		CONV$AB_OUT_FAB [ FAB$W_MRS ] = .COPY_FAB [ FAB$W_MRS ];	! Max record size
		CONV$AB_OUT_FAB [ FAB$L_MRN ] = .COPY_FAB [ FAB$L_MRN ];	! Max records
		CONV$AB_OUT_FAB [ FAB$W_BLS ] = .COPY_FAB [ FAB$W_BLS ];	! Block size
		CONV$AB_OUT_FAB [ FAB$B_BKS ] = .COPY_FAB [ FAB$B_BKS ];	! Bucket size
		CONV$AB_OUT_FAB [ FAB$B_FSZ ] = .COPY_FAB [ FAB$B_FSZ ];	! Fixed size
		CONV$AB_OUT_FAB [ FAB$W_GBC ] = .COPY_FAB [ FAB$W_GBC ];	! Global Buffers

		CONV$AB_OUT_FAB [ FAB$L_FOP ] = .CONV$AB_OUT_FAB [ FAB$L_FOP ] OR
						.COPY_FAB [ FAB$L_FOP ];	! File options

		! If the PROLOGUE option was specified and the file is indexed
		! then stuff the first key xab with the user value
		!
		IF ( .CONV$AB_OUT_FAB [ FAB$B_ORG ] EQLU FAB$C_IDX ) AND
						.CONV$AB_FLAGS [ CONV$V_PROLOG ]
		THEN
		    BEGIN

		    LOCAL	XAB : REF BLOCK [ ,BYTE ];

		    ! Find the first key xab
		    !
		    XAB = .CONV$AB_OUT_FAB [ FAB$L_XAB ];

		    ! The xabs have to be in order and there must be a key 0 so
		    ! the first one we find is the one we want
		    !
```

```
                        WHILE .XAB [ XAB$B_COD ] NEQU XAB$C_KEY
                        DO

                            ! If there are no more xabs then we really have a problem
                            ! so forget it
                            !
                            IF .XAB [ XAB$L_NXT ] EQLU 0
                            THEN
                                RETURN CONV$_BADLOGIC
                            ELSE
                                XAB = .XAB [ XAB$L_NXT ];

                        ! Stuff the value
                        !
                        XAB [ XAB$B_PROLOG ] = .CONV$GL_PROLOG

                        END;

                    ! Create it
                    !
                    ! If the record format was changed on a non VMS system
                    ! signal a warning (only to DCL)
                    !
                    $CREATE ( FAB=CONV$AB_OUT_FAB,ERR=CONV$$RMS_OPEN_ERROR );

                    IF ( $CREATE ( FAB=CONV$AB_OUT_FAB,ERR=CONV$$RMS_OPEN_ERROR ) EQLU
                                                                RMS$_CRE_STM ) AND
                                                    .CONV$AB_FLAGS [ CONV$V_DCL ]
                    THEN
                        SIGNAL ( CONV$_CREATEDSTM );

                    ! Since a create does not fill in the summary xab do a display
                    !
                    $DISPLAY( FAB=CONV$AB_OUT_FAB )

                    END
                ELSE
                    $OPEN ( FAB=CONV$AB_OUT_FAB,ERR=CONV$$RMS_OPEN_ERROR );

                ! If we got here then we have opened a file.
                !
                CONV$AB_FLAGS [ CONV$V_OUT ] = _SET;

                ! Set some bits depending on the type of output file
                !
                ! Can only append to a sequential file
                !
                IF .CONV$AB_OUT_FAB [ FAB$B_ORG ] EQLU FAB$C_SEQ
                THEN
                    CONV$AB_OUT_RAB [ RAB$V_EOF ] = .CONV$GL_APPEND
                ELSE

                    ! If append was on without a seq. output file then error
                    !
                    IF .CONV$GL_APPEND THEN RETURN CONV$_NOTSEQ;

                ! Is't not very exciting if it's not an index file
```

```
!
IF .CONV$AB_OUT_FAB [ FAB$B_ORG ] NEQU FAB$C_IDX
THEN
    BEGIN
    CONV$GL_MERGE = _CLEAR;
    CONV$GL_SORT  = _CLEAR;
    CONV$GL_FAST  = _CLEAR
    END
ELSE

    ! Set the fill option if it is indexed
    !
    CONV$AB_OUT_RAB [ RAB$V_LOA ] = NOT .CONV$GL_FILL;

! If we are sorting or fastloading
! then allocate space for KEY and AREA XAB's and fill them in by reading
! the prologue blocks in the file
!
IF ( .CONV$GL_FAST OR .CONV$GL_SORT )
THEN
    BEGIN

    ! Connect the file for Block IO for reading the
    ! prologue.
    !
    CONV$AB_OUT_RAB [ RAB$V_BIO ] = _SET;

    $CONNECT ( RAB=CONV$AB_OUT_RAB,ERR=CONV$$RMS_OPEN_ERROR );

    ! Read the prologue
    !
    CONV$$READ_PROLOGUE();

    ! If this is not a fast load then we need to bounce the file so we can
    ! do record IO again. (This sure doesen'd look good!)
    !
    IF NOT .CONV$GL_FAST
    THEN
        BEGIN

        ! Disconnect and Close (Dont check the disconnect)
        !
        $DISCONNECT ( RAB=CONV$AB_OUT_RAB );
        $CLOSE( FAB=CONV$AB_OUT_FAB );

        ! Clear the Block IO flag
        !
        CONV$AB_OUT_RAB [ RAB$V_BIO ] = _CLEAR;

        ! Reopen and Reconnect (Dont need to reconnect the PLG RAB)
        !
        $OPEN ( FAB=CONV$AB_OUT_FAB,ERR=CONV$$RMS_OPEN_ERROR );

        $CONNECT ( RAB=CONV$AB_OUT_RAB,ERR=CONV$$RMS_OPEN_ERROR )

        END
    END
```

```
2       ELSE
3           BEGIN
3
3           ! If we are merging into an indexed file
3           ! then set the access to KEY
3           !
3           IF .CONV$GL_MERGE
3           THEN
3               CONV$AB_OUT_RAB [ RAB$B_RAC ] = RAB$C_KEY;
3
3           ! If we are not sorting or fastloading
3           ! then connect the stream normally
3           !
3           $CONNECT ( RAB=CONV$AB_OUT_RAB,ERR=CONV$$RMS_OPEN_ERROR );
3
3           ! If the output file was not opened by now we can open it here
3           !
3           IF NOT .CONV$AB_FLAGS [ CONV$V_IN ]
3           THEN
4               RET_ON_ERROR( CONV$$OPEN_IN() )
4
2           END;
2
2       ! If PAD switch is on and the file is not fixed format
2       !
3       IF .CONV$GL_PAD AND ( .CONV$AB_OUT_FAB [ FAB$B_RFM ] NEQU FAB$C_FIX )
2       THEN
3           BEGIN
3           CONV$GL_PAD = _CLEAR;
3           SIGNAL( CONV$_PAD )
2           END;
2
2       ! Any errors on the output rab should be write errors (exceptions are in
2       ! the fast load code
2       !
2       CONV$AB_OUT_RAB [ RAB$L_CTX ] = CONV$_WRITEERR;
2
2       ! Return normally
2       !
2       RETURN CONV$_SUCCESS
2
1       END;


                                                   .EXTRN  SYS$PARSE, SYS$CREATE
                                                   .EXTRN  SYS$DISCONNECT, SYS$CLOSE

                               0FFC 00000          .ENTRY  CONV$$OPEN_OUTPUT, Save R2,R3,R4,R5,R6,R7,-  ; 0632
                                                           R8,R9,R10,R11                                 :
             59      0000G  CF  9E 00002           MOVAB   CONV$AB_FLAGS+2, R9                           :
             58      0000G  CF  9E 00007           MOVAB   CONV$GL_MERGE, R8                             :
             57  00000000G  00  9E 0000C           MOVAB   SYS$OPEN, R7                                  :
             56      0000G  CF  9E 00013           MOVAB   CONV$GL_FAST, R6                              :
             55  00000000G  00  9E 00018           MOVAB   SYS$CONNECT, R5                               :
             54      0000G  CF  9E 0001F           MOVAB   CONV$$RMS_OPEN_ERROR, R4                      :
             53      0000G  CF  9E 00024           MOVAB   CONV$AB_OUT_RAB+4, R3                         :
```

```
                                  52      0000G  CF  9E 00029        MOVAB   CONV$AB_OUT_FAB, R2                              :
                         18       A2  00000000G  8F  D0 0002E        MOVL    #CONV$_OPENOUT, CONV$AB_OUT_FAB+24               : 0690
                                  50      0000G  CF  D0 00036        MOVL    CONV$AR_OUT_FILE_NAM, R0                         : 0694
                         34       A2             60  90 0003B        MOVB    (R0), CONV$AB_OUT_FAB+52                         :
                         2C       A2         04  A0  D0 0003F        MOVL    4(R0), CONV$AB_OUT_FAB+44                        : 0695
                      0000G       CF      0000G  CF  90 00044        MOVB    CONV$AB_IN_FAB+52, CONV$AB_IN_NAM+3              : 0699
                      0000G       CF      0000G  CF  D0 0004B        MOVL    CONV$AB_IN_FAB+44, CONV$AB_IN_NAM+4              : 0700
                                                 14  BB 00052        PUSHR   #^M<R2,R4>                                       : 0704
                  00000000G       00             02  FB 00054        CALLS   #2, SYS$PARSE                                    :
                                  50         40  A2  D0 0005B        MOVL    CONV$AB_OUT_FAB+64, OUT_DEV                      : 0708
             02                   50             0D  E1 0005F        BBC     #13, OUT_DEV, 1$                                 : 0712
                                                 66  D4 00063        CLRL    CONV$GL_FAST                                     : 0714
05  A2       01                   01      0000G  CF  F0 00065 1$:    INSV    CONV$GL_WRITE_C, #1, #1, CONV$AB_OUT_FAB+5       : 0720
                                  50             68  D2 0006D        MCOML   CONV$GL_MERGE, R0                                : 0724
04  A2       01                   06             50  F0 00070        INSV    R0, #6, #1, CONV$AB_OUT_FAB+4                    :
                                  03      0000G  CF  E8 00076        BLBS    CONV$GL_CREATE, 2$                               : 0729
                                              0081   31 0007B        BRW     9$                                               :
                                  0D      0000G  CF  E9 0007E 2$:    BLBC    CONV$GL_FDL, 3$                                  : 0737
                                  50      0000'  CF  D0 00083        MOVL    CONV$AB_FDL_FAB, COPY_FAB                        : 0744
                      0000G       CF         24  A0  D0 00088        MOVL    36(COPY_FAB), CONV$AB_OUT_XABSUM+4               : 0748
                                                 0C  11 0008E        BRB     4$                                               :
                                  50      0000G  CF  9E 00090 3$:    MOVAB   CONV$AB_IN_FAB, COPY_FAB                         : 0757
                      0000G       CF      0000G  CF  D0 00095        MOVL    CONV$AB_IN_XABSUM+4, CONV$AB_OUT_XABSUM+4        : 0763
                         10       A2         10  A0  D0 0009C 4$:    MOVL    16(COPY_FAB), CONV$AB_OUT_FAB+16                 : 0769
                         14       A2         14  A0  B0 000A1        MOVW    20(COPY_FAB), CONV$AB_OUT_FAB+20                 : 0770
                         1C       A2         1C  A0  D0 000A6        MOVL    28(COPY_FAB), CONV$AB_OUT_FAB+28                 : 0771
                         36       A2         36  A0  B0 000AB        MOVW    54(COPY_FAB), CONV$AB_OUT_FAB+54                 : 0775
                         38       A2         38  A0  7D 000B0        MOVQ    56(COPY_FAB), CONV$AB_OUT_FAB+56                 : 0776
                         48       A2         48  A0  B0 000B5        MOVW    72(COPY_FAB), CONV$AB_OUT_FAB+72                 : 0780
                         04       A2         04  A0  C8 000BA        BISL2   4(COPY_FAB), CONV$AB_OUT_FAB+4                   : 0783
                                  20         1D  A2  91 000BF        CMPB    CONV$AB_OUT_FAB+29, #32                          : 0788
                                                 26  12 000C3        BNEQ    8$                                               :
             22                   69             06  E1 000C5        BBC     #6, CONV$AB_FLAGS+2, 8$                          : 0789
                                  50         24  A2  D0 000C9        MOVL    CONV$AB_OUT_FAB+36, XAB                          : 0797
                                  15             60  91 000CD 5$:    CMPB    (XAB), #21                                       : 0802
                                                 13  13 000D0        BEQL    7$                                               :
                                             04  A0  D5 000D2        TSTL    4(XAB)                                           : 0808
                                                 08  12 000D5        BNEQ    6$                                               :
                                  50  00000000G  8F  D0 000D7        MOVL    #CONV$_BADLOGIC, R0                              : 0810
                                                     04 000DE        RET                                                      :
                                  50         04  A0  D0 000DF 6$:    MOVL    4(XAB), XAB                                      : 0812
                                                 E8  11 000E3        BRB     5$                                               : 0808
                         48       A0      0000G  CF  90 000E5 7$:    MOVB    CONV$GL_PROLOG, 72(XAB)                          : 0816
                                                 14  BB 000EB 8$:    PUSHR   #^M<R2,R4>                                       : 0825
                  00000000G       00             02  FB 000ED        CALLS   #2, SYS$CREATE                                   :
                                                 52  DD 000F4        PUSHL   R2                                               : 0835
                  00000000G       00             01  FB 000F6        CALLS   #1, SYS$DISPLAY                                  :
                                                 05  11 000FD        BRB     10$                                              :
                                                 14  BB 000FF 9$:    PUSHR   #^M<R2,R4>                                       : 0839
                                  67             02  FB 00101        CALLS   #2, SYS$OPEN                                     :
                                  69             02  88 00104 10$:   BISB2   #2, CONV$AB_FLAGS+2                              : 0843
                                  50         1D  A2  9A 00107        MOVZBL  CONV$AB_OUT_FAB+29, R0                           : 0849
                                                 0A  12 0010B        BNEQ    11$                                              :
01  A3       01                   00      0000G  CF  F0 0010D        INSV    CONV$GL_APPEND, #0, #1, CONV$AB_OUT_RAB+5        : 0851
                                                 0D  11 00115        BRB     12$                                              :
                                  08      0000G  CF  E9 00117 11$:   BLBC    CONV$GL_APPEND, 12$                              : 0856
                                  50  00000000G  8F  D0 0011C        MOVL    #CONV$_NOTSEQ, R0                                :
```

```
                                                                  H  8
CONV$FILES      VAX-11 CONVERT                                    15-Sep-1984 23:45:35    VAX-11 Bliss-32 V4.0-742          Page  26
V04-000         OPEN_OUTPUT                                       14-Sep-1984 12:13:55    [CONV.SRC]CONVFILES.B32;1               (8)

                                                                04 00123          RET                                             ;
                                              20            50  91 00124 12$:    CMPB    R0, #32                                 ; 0860
                                                            0A  13 00127         BEQL    13$                                     ;
                                                            68  D4 00129         CLRL    CONV$GL_MERGE                           ; 0863
                                                      0000G CF  D4 0012B         CLRL    CONV$GL_SORT                            ; 0864
                                                            66  D4 0012F         CLRL    CONV$GL_FAST                            ; 0865
                                                            0B  11 00131         BRB     14$                                     ;
                                              50      0000G CF  D2 00133 13$:    MCOML   CONV$GL_FILL, R0                        ; 0871
     01   A3            01                    05            50  F0 00138         INSV    R0, #5, #1, CONV$AB_OUT_RAB+5           ;
                                              05            66  E8 0013E 14$:    BLBS    CONV$GL_FAST, 15$                       ; 0877
                                              38      0000G CF  E9 00141         BLBC    CONV$GL_SORT, 16$                       ;
                                          01  A3            08  88 00146 15$:    BISB2   #8, CONV$AB_OUT_RAB+5                   ; 0884
                                                            54  DD 0014A         PUSHL   R4                                      ; 0886
                                                         FC A3  9F 0014C         PUSHAB  CONV$AB_OUT_RAB                         ;
                                              65            02  FB 0014F         CALLS   #2, SYS$CONNECT                         ;
                                                          0000G 30 00152         BSBW    CONV$$READ_PROLOGUE                     ; 0890
                                              40            66  E8 00155         BLBS    CONV$GL_FAST, 18$                       ; 0895
                                                         FC A3  9F 00158         PUSHAB  CONV$AB_OUT_RAB                         ; 0901
                                  000000000G  00            01  FB 0015B         CALLS   #1, SYS$DISCONNECT                      ;
                                                            52  DD 00162         PUSHL   R2                                      ; 0902
                                  000000000G  00            01  FB 00164         CALLS   #1, SYS$CLOSE                           ;
                                          01  A3            08  8A 0016B         BICB2   #8, CONV$AB_OUT_RAB+5                   ; 0906
                                                            14  BB 0016F         PUSHR   #^M<R2,R4>                              ; 0910
                                              67            02  FB 00171         CALLS   #2, SYS$OPEN                            ;
                                                            54  DD 00174         PUSHL   R4                                      ; 0912
                                                         FC A3  9F 00176         PUSHAB  CONV$AB_OUT_RAB                         ;
                                              65            02  FB 00179         CALLS   #2, SYS$CONNECT                         ;
                                                            1A  11 0017C         BRB     18$                                     ; 0895
                                              04            68  E9 0017E 16$:    BLBC    CONV$GL_MERGE, 17$                      ; 0922
                                          1A  A3            01  90 00181         MOVB    #1, CONV$AB_OUT_RAB+30                  ; 0924
                                                            54  DD 00185 17$:    PUSHL   R4                                      ; 0929
                                                         FC A3  9F 00187         PUSHAB  CONV$AB_OUT_RAB                         ;
                                              65            02  FB 0018A         CALLS   #2, SYS$CONNECT                         ;
                                              08            69  E8 0018D         BLBS    CONV$AB_FLAGS+2, 18$                    ; 0933
                                        FD56  CF            00  FB 00190         CALLS   #0, CONV$$OPEN_IN                       ; 0935
                                              27            50  E9 00195         BLBC    STATUS, 20$                             ;
                                              17      0000G CF  E9 00198 18$:    BLBC    CONV$GL_PAD, 19$                        ; 0941
                                              01         1F A2  91 0019D         CMPB    CONV$AB_OUT_FAB+31, #1                  ;
                                                            11  13 001A1         BEQL    19$                                     ;
                                                      0000G CF  D4 001A3         CLRL    CONV$GL_PAD                             ; 0944
                                                 000000000G 8F  DD 001A7         PUSHL   #CONV$_PAD                              ; 0945
                                  000000000G  00            01  FB 001AD         CALLS   #1, LIB$SIGNAL                          ;
                                          14  A3 000000000G 8F  D0 001B4 19$:    MOVL    #CONV$_WRITEERR, CONV$AB_OUT_RAB+24     ; 0951
                                              50            01  D0 001BC         MOVL    #1, R0                                  ; 0955
                                                                04 001BF 20$:    RET                                             ; 0957

; Routine Size:  448 bytes,    Routine Base:  _CONV$CODE + 0210
```

```
1 %SBTTL  'CREATE_BUFFER'
1 GLOBAL ROUTINE  CONV$$CREATE_BUFFER =
1 !++
1 !
1 ! Functional Description:
1 !
1 !       Creates the main record buffer and sets the record sizes
1 !
1 !       The main record buffer:
1 !
1 !               -------------------------------------//-----------------
1 !               |               |                                      |
1 !               |               |                                      |
1 !               -------------------------------------//-----------------
1 !               ^               ^
1 !               |               |
1 !           VFC_BUF_PTR     REC_BUF_PTR
1 !
1 !
1 ! Calling Sequence:
1 !
1 !       CONV$$CREATE_BUFFER()
1 !
1 ! Input Parameters:
1 !       none
1 !
1 ! Implicit Inputs:
1 !       none
1 !
1 ! Output Parameters:
1 !       none
1 !
1 ! Implicit Outputs:
1 !
1 !       Record buffer pointers and size variables
1 !
1 ! Routine Value:
1 !
1 !       SS$_NORMAL
1 !
1 ! Routines Called:
1 !
1 !       CONV$$GET_VM
1 !
1 ! Side Effects:
1 !       none
1 !
1 !--
1
2    BEGIN
2
2    LITERAL
2       MAX_REC_CTRL    = 14;   ! Maximun number of control bytes for a data
2                               ! record in index file (14 is for a fully
2                               ! compressed prologue 3 record
2
2    LOCAL
```

```
2              IN_VFC,
2              IN_MRS,
2              OUT_VFC,
2              OUT_EXTRA;
2
2          ! Acccount for the VFC temporaraly
2          !
2          IF .CONV$AB_OUT_FAB [ FAB$B_RFM ] EQL FAB$C_VFC
2          THEN
2              OUT_VFC = .CONV$AB_OUT_FAB [ FAB$B_FSZ ]
2          ELSE
2              OUT_VFC = 0;
2
2          ! If output MRS = 0 ( ie. VAR and VFC records ) then
2          ! check for Block Spanning with Sequential Files
2          ! and Bucket Crossing with Relative and Indexed
2          !
2          IF ( CONV$GW_OUT_MRS = .CONV$AB_OUT_FAB [ FAB$W_MRS ] ) EQL 0
2          THEN
3              BEGIN
3
3              LOCAL   OUT_DEV : BLOCK [ 1,LONG ];
3
3              ! Find out if this thing is going to tape, if so use block size
3              ! (Since records cannot spand blocks on tape)
3              !
3              OUT_DEV = .CONV$AB_OUT_FAB [ FAB$L_DEV ];
3
3              IF .OUT_DEV [ DEV$V_SQD ]
3              THEN
3                  CONV$GW_OUT_MRS = .CONV$AB_OUT_FAB [ FAB$W_BLS ] - .OUT_VFC - 2
3
3              ! Sequential and  NO Block spanning
3              !
3              ELSE IF ( .CONV$AB_OUT_FAB [ FAB$B_ORG ] EQLU FAB$C_SEQ ) AND
3                                                  .CONV$AB_OUT_FAB [ FAB$V_BLK ]
3              THEN
3                  CONV$GW_OUT_MRS = BLOCK_SIZE - .OUT_VFC - 2
3
3              ! Relative
3              !
3              ELSE IF .CONV$AB_OUT_FAB [ FAB$B_ORG ] EQLU FAB$C_REL
3              THEN
3                  CONV$GW_OUT_MRS = ( .CONV$AB_OUT_FAB [ FAB$B_BKS ] * BLOCK_SIZE ) -
3                                                                  .OUT_VFC - 3
3
3              ! Indexed
3              !
3              ELSE
3                  CONV$GW_OUT_MRS = ( .CONV$AB_OUT_FAB [ FAB$B_BKS ] * BLOCK_SIZE ) -
3                                                                  .OUT_VFC - 7;
3
2              END;
2
2          ! If the Input File is UDF then the UDF_MRS is calculated from
2          ! the output file attributes
2          !
```

```
	IF .CONV$AB_IN_FAB [ FAB$B_RFM ] EQLU FAB$C_UDF
	THEN
	    BEGIN

	    IN_MRS = BLOCK_SIZE;

	    ! If fixed format then no problem use that value, if
	    ! not see if a 512 byte record will fit
	    !
	    IF .CONV$AB_OUT_FAB [ FAB$B_RFM ] EQL FAB$C_FIX
	    THEN
		CONV$GW_UDF_MRS = .CONV$AB_OUT_FAB [ FAB$W_MRS ]
	    ELSE

		! If the udf record will not if into the output file then error
		!
		IF .CONV$GW_OUT_MRS LSS BLOCK_SIZE
		THEN
		    RETURN CONV$_UDF_BLK
		ELSE
		    CONV$GW_UDF_MRS = BLOCK_SIZE
	    END
	ELSE
	    BEGIN

	    ! Here for a normal input file
	    ! IN_MRS is the length of the maximun record size
	    !
	    ! Now see if the file is VFC
	    !
	    IF .CONV$AB_IN_FAB [ FAB$B_RFM ] EQL FAB$C_VFC
	    THEN
		IN_VFC = .CONV$AB_IN_FAB [ FAB$B_FSZ ]
	    ELSE
		IN_VFC = 0;

	    ! If max. record size is zero then we find out from Longest Record Length
	    ! on disk or Block Size for magtape
	    !
	    IF ( IN_MRS = .CONV$AB_IN_FAB [ FAB$W_MRS ] ) EQL 0
	    THEN
		BEGIN

		LOCAL	IN_DEV : BLOCK [ 1,LONG ];

		! Find out if this thing is comming from tape if so use block size
		! (Since records cannot spand blocks on tape)
		!
		IN_DEV = .CONV$AB_IN_FAB [ FAB$L_DEV ];

		IF .IN_DEV [ DEV$V_SQD ]
		THEN
		    IN_MRS = .CONV$AB_IN_FAB [ FAB$W_BLS ] - .IN_VFC

		! If SEQ use LRL otherwise check
		! bucket sizes
		!
```

```
                ELSE IF .CONV$AB_IN_FAB [ FAB$B_ORG ] EQL FAB$C_SEQ
                THEN
                    IN_MRS = .CONV$AB_IN_XABFHC [ XAB$W_LRL ]

                ! Relative
                !
                ELSE IF .CONV$AB_IN_FAB [ FAB$B_ORG ] EQL FAB$C_REL
                THEN
                    IN_MRS = ( .CONV$AB_IN_FAB [ FAB$B_BKS ] * BLOCK_SIZE ) -
                                                                .IN_VFC - 3

                ! Indexed
                !
                ELSE
                    IN_MRS = ( .CONV$AB_IN_FAB [ FAB$B_BKS ] * BLOCK_SIZE ) -
                                                                .IN_VFC - 7

                END

            END;

        ! Now calculate the number of blocks needed.
        !
        ! If UDF, ask for one block extra for overlapping of the buffers
        !
        IF .CONV$AB_IN_FAB [ FAB$B_RFM ] EQLU FAB$C_UDF
        THEN
            OUT_EXTRA = BLOCK_SIZE
        ELSE
            OUT_EXTRA = 0;

        BEGIN

        LOCAL
            BYTES,
            VFC_OFFSET;

        ! Determine which is larger and use that size for the Buffer Size
        !
        BYTES = MAX( BLOCK_SIZE ,                                      ! At least a page
                    ( .IN_MRS + .IN_VFC ),                             ! Input record size
                    ( .CONV$GW_OUT_MRS + .OUT_VFC + .OUT_EXTRA )); ! Output record size

        ! If we are doing a fast load get some extra bytes to use at the beginning
        ! of the record for control information
        !
        IF .CONV$GL_FAST
        THEN
            BYTES = .BYTES + MAX_REC_CTRL;

        !
        ! IF UDF input, round buffer up to next whole block
        !
        IF .CONV$AB_IN_FAB [ FAB$B_RFM ] EQLU FAB$C_UDF
        THEN
            BYTES = (.BYTES + 511) AND NOT 511;
```

```
; 1194          1186  3        ! Create the Buffer from virtural memory
; 1195          1187  3        !
; 1196          1188  3        CONV$GL_REC_BUF_PTR = CONV$$GET_VM ( .BYTES );
; 1197          1189  3
; 1198          1190  3        ! If we doing a fast load hide the extra bytes at the beginning of
; 1199          1191  3        ! record.
; 1200          1192  3        !
; 1201          1193  3        IF .CONV$GL_FAST
; 1202          1194  3        THEN
; 1203          1195  3            CONV$GL_REC_BUF_PTR = .CONV$GL_REC_BUF_PTR + MAX_REC_CTRL;
; 1204          1196  3
; 1205          1197  3        ! Set the VFC offset to the max of the two offsets
; 1206          1198  3        !
; 1207          1199  3        VFC_OFFSET = MAX( .IN_VFC,.OUT_VFC );
; 1208          1200  3
; 1209          1201  3        ! Correct the pointers and set the max. record size
; 1210          1202  3        !
; 1211          1203  3        CONV$GL_VFC_BUF_PTR = .CONV$GL_REC_BUF_PTR;
; 1212          1204  3        CONV$GL_REC_BUF_PTR = .CONV$GL_VFC_BUF_PTR + .VFC_OFFSET;
; 1213          1205  3
; 1214          1206  3        CONV$GW_MAX_REC_SIZ = .BYTES - .VFC_OFFSET
; 1215          1207  3
; 1216          1208  2        END;
; 1217          1209  2
; 1218          1210  2        RETURN CONV$_SUCCESS
; 1219          1211  2
; 1220          1212  1        END;



                                                 0FFC 00000        .ENTRY  CONV$$CREATE_BUFFER, Save R2,R3,R4,R5,R6,-   ; 0959
                                                                           R7,R8,R9,R10,R11                             ;
                          59      0000G   CF  9E 00002             MOVAB   CONV$GL_REC_BUF_PTR, R9                      ;
                          58      0000G   CF  9E 00007             MOVAB   CONV$AB_IN_FAB+31, R8                        ;
                          57      0000G   CF  9E 0000C             MOVAB   CONV$GW_OUT_MRS, R7                          ;
                          56      0000G   CF  9E 00011             MOVAB   CONV$AB_OUT_FAB+31, R6                       ;
                          03              66  91 00016             CMPB    CONV$AB_OUT_FAB+31, #3                       ; 1022
                                          06  12 00019             BNEQ    1$                                           ;
                          54         20   A6  9A 0001B             MOVZBL  CONV$AB_OUT_FAB+63, OUT_VFC                  ; 1024
                                          02  11 0001F             BRB     2$                                           ;
                                          54  D4 00021 1$:         CLRL    OUT_VFC                                      ; 1026
                          67         17   A6  B0 00023 2$:         MOVW    CONV$AB_OUT_FAB+54, CONV$GW_OUT_MRS          ; 1032
                                          42  12 00027             BNEQ    6$                                           ;
                          50         21   A6  D0 00029             MOVL    CONV$AB_OUT_FAB+64, OUT_DEV                  ; 1041
              0D          50              05  E1 0002D             BBC     #5, OUT_DEV, 3$                              ; 1043
                          50         1D   A6  3C 00031             MOVZWL  CONV$AB_OUT_FAB+60, R0                       ; 1045
                          50              54  C2 00035             SUBL2   OUT_VFC, R0                                  ;
              67          50              02  A3 00038             SUBW3   #2, R0, CONV$GW_OUT_MRS                      ;
                                          2D  11 0003C             BRB     6$                                           ;
                          51         FE   A6  9A 0003E 3$:         MOVZBL  CONV$AB_OUT_FAB+29, R1                       ; 1049
                                          0D  12 00042             BNEQ    4$                                           ;
              08       FF  A6             03  E1 00044             BBC     #3, CONV$AB_OUT_FAB+30, 4$                   ; 1050
              67     01FE  8F             54  A3 00049             SUBW3   OUT_VFC, #510, CONV$GW_OUT_MRS               ; 1052
                                          1A  11 0004F             BRB     6$                                           ;
                          50         1F   A6  9A 00051 4$:         MOVZBL  CONV$AB_OUT_FAB+62, R0                       ; 1058
```

```
                                                         N  8
CONV$FILES       VAX-11 CONVERT                     15-Sep-1984 23:45:35    VAX-11 Bliss-32 V4.0-742          Page  32
V04-000          CREATE_BUFFER                      14-Sep-1984 12:13:55    [CONV.SRC]CONVFILES.B32;1               (9)

   50               50               09  78 00055          ASHL    #9, R0, R0                              ;
                    50               54  C2 00059          SUBL2   OUT_VFC, R0                             ; 1059
                    10               51  91 0005C          CMPB    R1, #16                                 ; 1056
                                     06  12 0005F          BNEQ    5$                                      ;
   67               50               03  A3 00061          SUBW3   #3, R0, CONV$GW_OUT_MRS                 ; 1059
                                     04  11 00065          BRB     6$                                      ; 1058
   67               50               07  A3 00067 5$:      SUBW3   #7, R0, CONV$GW_OUT_MRS                 ; 1065
                    51               68  9A 0006B 6$:      MOVZBL  CONV$AB_IN_FAB+31, R1                   ; 1072
                                     55  D4 0006E          CLRL    R5                                      ;
                                     51  D5 00070          TSTL    R1                                      ;
                                     2C  12 00072          BNEQ    9$                                      ;
                                     55  D6 00074          INCL    R5                                      ;
                    50        0200   8F  3C 00076          MOVZWL  #512, IN_MRS                            ; 1076
                    01               66  91 0007B          CMPB    CONV$AB_OUT_FAB+31, #1                  ; 1081
                                     08  12 0007E          BNEQ    7$                                      ;
           0000G    CF          17   A6  B0 00080          MOVW    CONV$AB_OUT_FAB+54, CONV$GW_UDF_MRS     ; 1083
                                     63  11 00086          BRB     15$                                     ;
           0200     8F               67  B1 00088 7$:      CMPW    CONV$GW_OUT_MRS, #512                   ; 1088
                                     08  1E 0008D          BGEQU   8$                                      ;
                    50  00000000G    8F  D0 0008F          MOVL    #CONV$_UDF_BLK, R0                      ; 1090
                                         04 00096          RET                                             ;
           0000G    CF        0200   8F  B0 00097 8$:      MOVW    #512, CONV$GW_UDF_MRS                   ; 1092
                                     4B  11 0009E          BRB     15$                                     ; 1081
                    03               51  91 000A0 9$:      CMPB    R1, #3                                  ; 1102
                                     06  12 000A3          BNEQ    10$                                     ;
                    52          20   A8  9A 000A5          MOVZBL  CONV$AB_IN_FAB+63, IN_VFC               ; 1104
                                     02  11 000A9          BRB     11$                                     ;
                                     52  D4 000AB 10$:     CLRL    IN_VFC                                  ; 1106
                    50          17   A8  3C 000AD 11$:     MOVZWL  CONV$AB_IN_FAB+54, IN_MRS               ; 1111
                                     38  12 000B1          BNEQ    15$                                     ;
                    51          21   A8  D0 000B3          MOVL    CONV$AB_IN_FAB+64, IN_DEV               ; 1120
   09               51               05  E1 000B7          BBC     #5, IN_DEV, 12$                         ; 1122
                    50          1D   A8  3C 000BB          MOVZWL  CONV$AB_IN_FAB+60, IN_MRS               ; 1124
                    50               52  C2 000BF          SUBL2   IN_VFC, IN_MRS                          ;
                                     27  11 000C2          BRB     15$                                     ;
                    53          FE   A8  9A 000C4 12$:     MOVZBL  CONV$AB_IN_FAB+29, R3                   ; 1129
                                     07  12 000C8          BNEQ    13$                                     ;
                    50       0000G   CF  3C 000CA          MOVZWL  CONV$AB_IN_XABFHC+10, IN_MRS            ; 1131
                                     1A  11 000CF          BRB     15$                                     ;
                    51          1F   A8  9A 000D1 13$:     MOVZBL  CONV$AB_IN_FAB+62, R1                   ; 1137
   51               51               09  78 000D5          ASHL    #9, R1, R1                              ;
                    51               52  C2 000D9          SUBL2   IN_VFC, R1                              ; 1138
                    10               53  91 000DC          CMPB    R3, #16                                 ; 1135
                                     06  12 000DF          BNEQ    14$                                     ;
                    50          FD   A1  9E 000E1          MOVAB   -3(R1), IN_MRS                          ; 1138
                                     04  11 000E5          BRB     15$                                     ; 1137
                    50          F9   A1  9E 000E7 14$:     MOVAB   -7(R1), IN_MRS                          ; 1144
                    07               55  E9 000EB 15$:     BLBC    R5, 16$                                 ; 1154
                    53        0200   8F  3C 000EE          MOVZWL  #512, OUT_EXTRA                         ; 1156
                                     02  11 000F3          BRB     17$                                     ;
                                     53  D4 000F5 16$:     CLRL    OUT_EXTRA                               ; 1158
                    50               52  C0 000F7 17$:     ADDL2   IN_VFC, R0                              ; 1169
                    51               67  3C 000FA          MOVZWL  CONV$GW_OUT_MRS, R1                     ; 1170
                    51               54  C0 000FD          ADDL2   OUT_VFC, R1                             ;
                    51               53  C0 00100          ADDL2   OUT_EXTRA, R1                           ;
                    51               50  D1 00103          CMPL    R0, R1                                  ;
                                     03  18 00106          BGEQ    18$                                     ;
```

```
                              50             51 D0 00108          MOVL    R1, R0                                   ;
                    00000200  8F             50 D1 0010B 18$:     CMPL    R0, #512                                 ; 1168
                              05             18 00112             BGEQ    19$                                      ;
                              50       0200  8F 3C 00114          MOVZWL  #512, R0                                 ;
                              51             50 D0 00119 19$:     MOVL    R0, BYTES                                ;
                              03      0000G  CF E9 0011C          BLBC    CONV$GL_FAST, 20$                        ; 1175
                              51             0E C0 00121          ADDL2   #14, BYTES                               ; 1177
                              0D             55 E9 00124 20$:     BLBC    R5, 21$                                  ; 1182
                              50       01FF  C1 9E 00127          MOVAB   511(R1), R0                              ; 1184
                51            50 000001FF    8F CB 0012C          BICL3   #511, R0, BYTES                          ;
                                             51 DD 00134 21$:     PUSHL   BYTES                                    ; 1188
                                        0000G 30 00136            BSBW    CONV$$GET_VM                             ;
                              5E             04 C0 00139          ADDL2   #4, SP                                   ;
                              69             50 D0 0013C          MOVL    R0, CONV$GL_REC_BUF_PTR                  ;
                              03      0000G  CF E9 0013F          BLBC    CONV$GL_FAST, 22$                        ; 1193
                              69             0E C0 00144          ADDL2   #14, CONV$GL_REC_BUF_PTR                 ; 1195
                              54             52 D1 00147 22$:     CMPL    R2, OUT_VFC                              ; 1199
                              03             18 0014A             BGEQ    23$                                      ;
                              52             54 D0 0014C          MOVL    OUT_VFC, R2                              ;
                              50             52 D0 0014F 23$:     MOVL    R2, VFC_OFFSET                           ;
                       0000G  CF             69 D0 00152          MOVL    CONV$GL_REC_BUF_PTR, CONV$GL_VFC_BUF_PTR ; 1203
                              69   0000GDF40 9E 00157             MOVAB   @CONV$GL_VFC_BUF_PTR[VFC_OFFSET], -      ; 1204
                                                                          CONV$GL_REC_BUF_PTR                      ;
          0000G  CF           51             50 A3 0015D          SUBW3   VFC_OFFSET, BYTES, CONV$GW_MAX_REC_SIZ   ; 1206
                              50             01 D0 00163          MOVL    #1, R0                                   ; 1210
                                                04 00166          RET                                              ; 1212

; Routine Size:  359 bytes,    Routine Base:  _CONV$CODE + 03D0


; 1221          1213  1
; 1222          1214  0 END      ELUDOM



                                                                  .EXTRN  LIB$SIGNAL

;                              PSECT SUMMARY
;
;       Name                   Bytes                    Attributes
;
;  _CONV$GLOBAL                      8  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,  PIC,ALIGN(2)
;  _CONV$CODE                     1335  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(2)



;                          Library Statistics
;
;                                            -------- Symbols --------       Pages       Processing
;       File                                 Total     Loaded  Percent       Mapped      Time
;
;  _$255$DUA28:[SYSLIB]LIB.L32;1             18619        96         0        1000         00:01.7
;  _$255$DUA28:[CONV.SRC]CONVERT.L32;1         165        14         8          17         00:00.2
;                                   COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:CONVFILES/OBJ=OBJ$:CONVFILES MSRC$:CONVFILES/UPDATE=(ENH$:CONVFILES)

; Size:          1335 code + 8 data bytes
; Run Time:         00:29.5
; Elapsed Time:     01:20.5
; Lines/CPU Min:     2473
; Lexemes/CPU-Min: 18442
; Memory Used:  204 pages
; Compilation Complete
```

0065 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

CONVDATA
LIS

CONVFSTLD
LIS

CONVFSTIO
LIS

CONVFSTRC
LIS

CONVFILES
LIS

CONVERROR
LIS

CONVFASTM
LIS

CONVDCL
LIS